LaTiS: A Data Server to Address Data Interoperability

Anne Wilson, Doug Lindholm

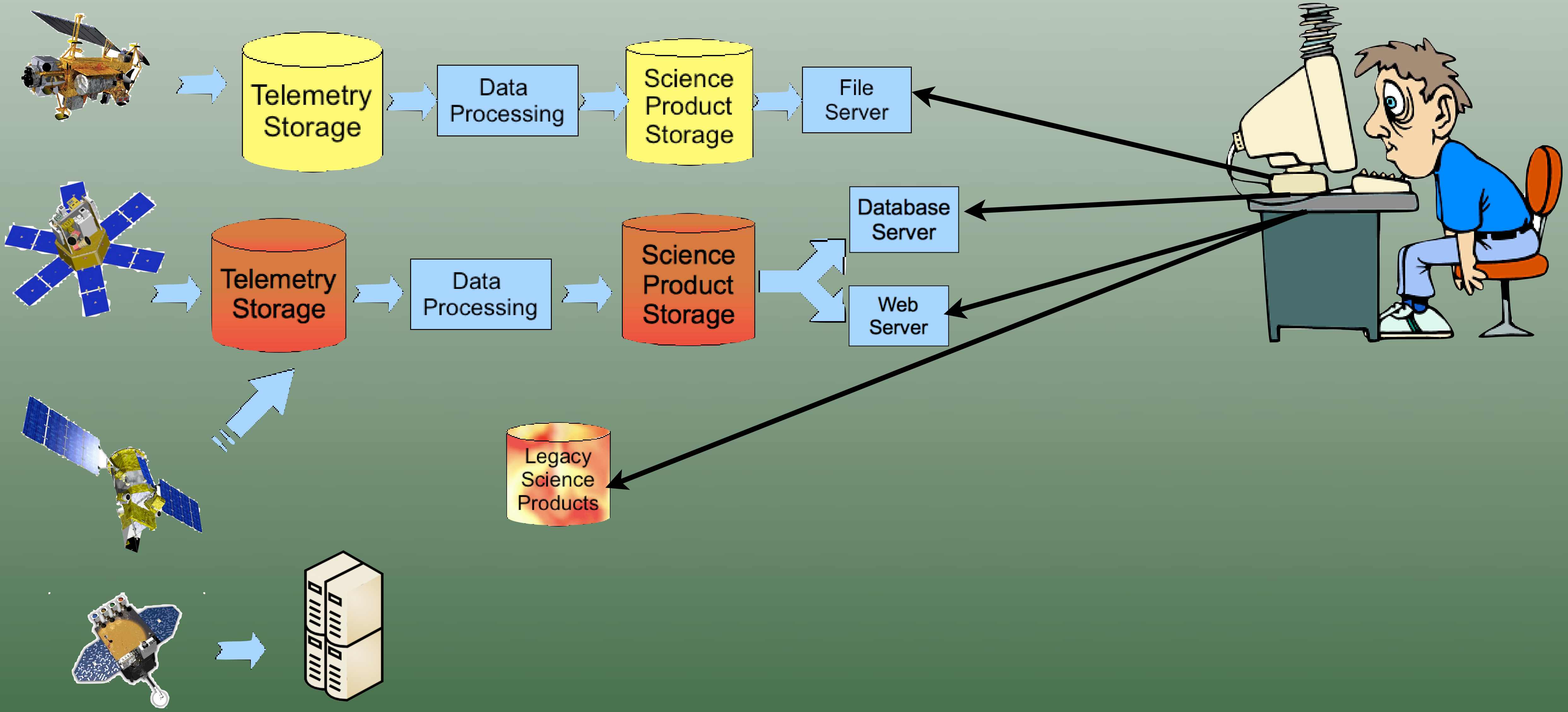
Laboratory for Atmospheric and Space Physics
University of Colorado, Boulder
anne.wilson@lasp.colorado.edu
doug.lindholm@lasp.colorado.edu

ESDSWG Annual Meeting
October 21, 2010
New Orleans

The Data Interoperability Problem

- Data formats vary significantly within and across scientific domains
- Multiple competing standards
- Standard formats are used without following conventions
- Home grown data formats

- Data users spend time and resources acquiring and formatting data for their needs
- Data providers spend time and resources reformatting their data for broader user access



The Problem with File Oriented Data Access

- Users request and receive collections of files that they must manage
- Files may not contain precisely what is desired, generally subsequent processing is required, particularly reformatting
- File based FTP is commonly relied upon
- File centricity often requires knowing file names a priori
- File oriented abstractions, not data oriented

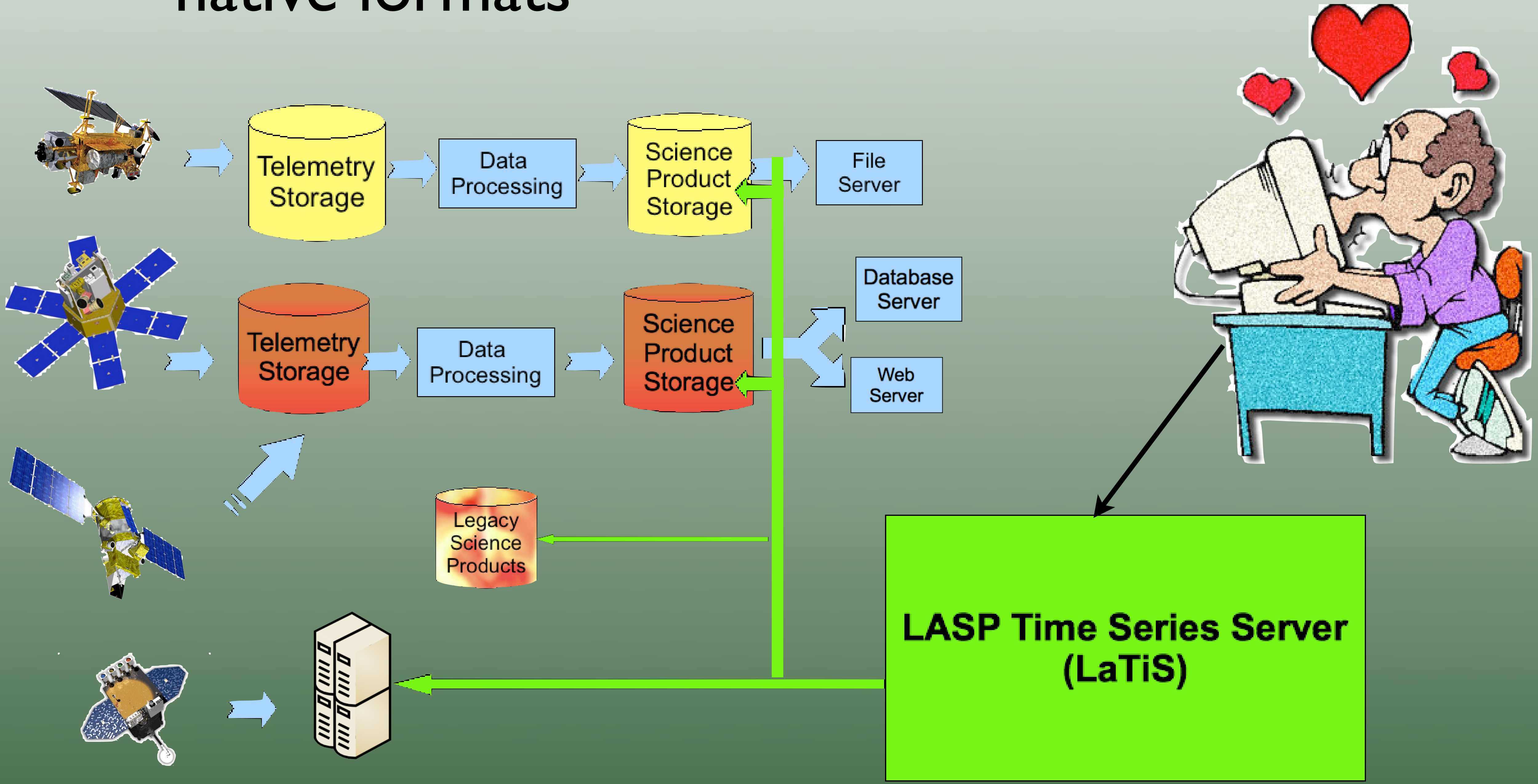
LaTiS: A Data Server for Time Based Data

- LASP Time Series Server
  - Early version known as TSDS, Time Series Data Server
    - Used operationally in LISIRD web app: http://lasp.colorado.edu/lisird/
    - http://sourceforge.net/projects/tsds/
- Supports data interoperability via an extensible architecture that can read and write data in a variety of formats
- Provides high level abstractions for data access
  - Not file centric, e.g., can serve from a RDBMS
- Open source, Java, JEE compatible, easy to install

Pluggable Architecture Supports Extensions to Handle New Data Formats

- Reads/Writes data in a variety of input/output formats via Reader/Writer Plugins
  - If a plugin exists to read a particular format, that code can be reused for subsequent datasets of the same format
  - If a plugin does not exist for a particular format, a new one can be added, supported by the pluggable architecture
- Provides a uniform interface to datasets
- Uses info in TSML descriptor file
- Vision: library of Reader and Writer plugins, including community contributions

- Data users can use data without processing
- Data providers can serve data in their native formats



High Level Abstractions for Data Access

- Serves data as a function of time
- Supports subsetting, filtering, (soon) transforming, aggregation
- Emphasis on data access
  - Provides support for (does not preclude) discovery, provenance
- OPeNDAP compliant, and more
  - Independent, lightweight implementation of DAP2 spec
  - Interface extended to handle additional functionality
- RESTful API
  - http://**host**/latis/**dataset.suffix**?**constraint_expression**
  - e.g., http://lasp.colorado.edu/lisird/tss/sorce_tsi_24hr.csv?time,tsi_1au&format_time(yyyy-DDD)&time>2010-01-01

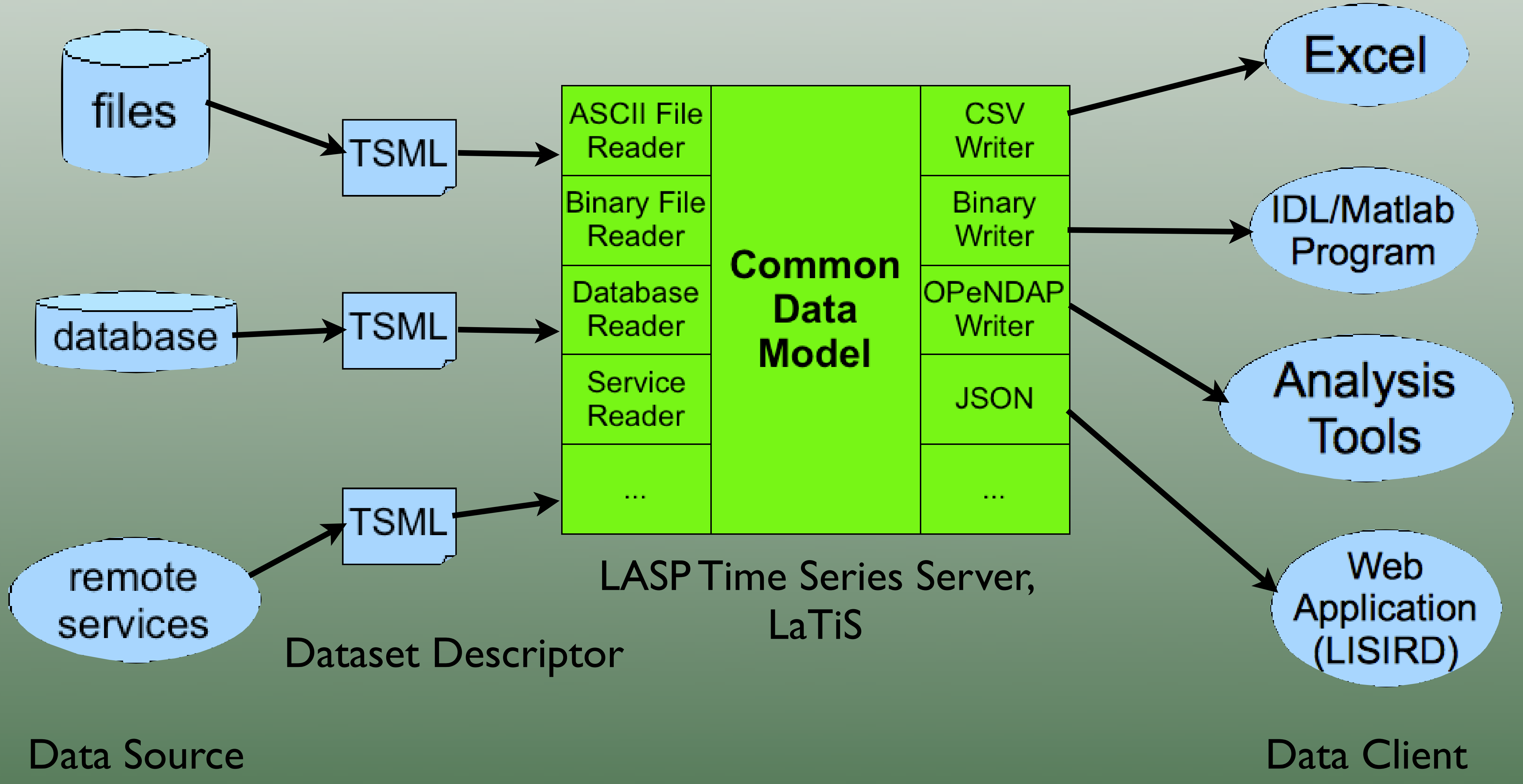
Installation

- .jar file for inclusion in a web app
- .war file to run as separate web app
- Drop jar or war file in appropriate location
- Create TSML descriptor file for each dataset to be served
- Add entry for each dataset in THREDDS catalog

Uses a Common Data Model

- A format agnostic representation of a dataset
  - Can support slicing, dicing, subsetting, filtering, aggregating, transformation
- Based on Unidata Common Data Model (CDM)
  - Merge of NetCDF Classic, HDF5, OPeNDAP data models
- Modified to meet needs revealed by application in new domain of space physics

Interoperability via a Common Data Model



LaTiS Data Model

- Inspired by Unidata CDM, with different semantics
- Object oriented over Array based
- Functional relationship: data as a function of time
  - Independent, dependent variable concept simplifies code
- Data storage agnostic, beyond file abstraction
- Virtual dataset: subset and filter before reading data
- Implementation independent API
  - We provide Java, (soon) IDL implementation
  - Others can create implementations in their favorite language
- Extensible with custom variable types as plugins
  - E.g., "Spectrum" can be defined as desired
    - For space physics, "Spectrum" would use "wavelength" as independent variable

TSML: Time Series Markup Language

- Based on Unidata NcML
  - Unidata goal is to “enhance NetCDF file”
- We evolved TSML to:
  - support non self describing formats
  - support data reshaping, e.g., drop a variable
  - remove file centricity
  - Provide better info for serving the data, e.g., Reader parameters

```
<variable name=”TimeSeries”>
  <dimension name="time"/>
  <variable name=”time”/>
  <variable name=”spectrum”>
    <dimension name=”wavelength” length=”100”/>
    <variable name=”wavelength”/>
    <variable name=”a”/>
  </variable>
</variable>
```

Discovery Metadata Maintained in a THREDDS Catalog

- May experiment with different implementation to better support search

LaTiS Roadmap

- HDF plugins
- Other formats, filters
- LaTiS available by December, 2010, Fall AGU
- Extend beyond time series abstraction
    - Geolocated data
- Data storage in the cloud